Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) A method of logging updates <u>and recovering from failure</u> in a ~~main-memory~~-transaction-processing system having main memory for storing a database, one or more ~~log disks~~<u>persistent backup storage devices</u> for storing <u>the database in the main memory</u>~~log records for parallel recovery of the main-memory database~~, and one or more ~~backup disks~~<u>persistent log storage devices</u> for storing ~~a copy of the main-memory database, the method comprising the steps of~~<u>log records for parallel logging and parallel recovery, the logging method</u> <u>comprising</u>:

        ~~taking a before-image of the database before an update to the database is made;~~

        ~~taking an after-image of the database after the update is made;~~

        ~~generating~~ a differential log <u>records</u> ~~as a log body of each log record~~ by applying ~~a~~ bit-wise exclusive-OR (XOR) operation between ~~the~~<u>a</u> before-<u>update</u> image and ~~the~~<u>an</u> after-<u>update</u> image;

        <u>distributing the generated differential log records in parallel to said persistent log storage</u> <u>devices; and</u>

————recovering from a failure by ~~applying~~ <u>replaying the differential records in an</u> <u>arbitrary order, which is independent of the order of generation of the log records, by using</u> <u>bit-wise</u> ~~the~~ XOR operation<u>s.</u> ~~between the differential log and the before-image.~~

2.   (Original) The method of Claim 1, wherein the database comprises a plurality of fixed-size pages.

3.   (Original) The method of Claim 2, wherein said each log record has a log header comprising:

LSN (Log Sequence Number) for storing log sequence;

TID(Transaction ID) for storing the identity of the transaction that created the log record;

Previous LSN for storing the identity of the most recently created log by the same transaction;

Type for storing the type of the log record;

Backup ID for storing the relation between the log record and the updated page for use with fuzzy checkpointing;

Page ID for storing the identity of an updated page;

Offset for storing the starting offset of an updated area within the updated page; and

Size for storing the size of the updated area.

4.   (Currently Amended) The method of Claim 1, further comprising the step of:

checkpointing by occasionally writing the database in the main memory to said one or

more backup disks as backup datapersistent back storage devices.

5.   (Currently Amended) The method of Claim 4, wherein the step of checkpointing uses

the a transaction consistent checkpointing policy.

6.   (Currently Amended) The method of Claim 4, wherein the step of checkpointing uses

the an action consistent checkpointing policy.

7.   (Currently Amended) The method of Claim 4, wherein the step of checkpointing uses

the a fuzzy checkpointing policy.

8.   (Currently Amended) The method of Claim 4, wherein the step of recovering

comprises the steps of:

loading the backup data checkpointed database from said one or more backup

diskspersistent backup storage devices into the main memory database; and

_____——loading the log from said one or more ~~log disks~~<u>persistent log storage devices</u> into

the main memory database in order to restore the main memory database to the most recent

consistent state.

9.   (Currently Amended) The method of Claim 8, wherein ~~the step of~~ loading the ~~backup~~

~~data~~<u>checkpointed database</u> is executed in parallel by partitioning the backup data.

10.   (Currently Amended) The method of Claim 8, wherein <u>replaying the log records is</u>

<u>done in two passes by separating a redoing pass and an undoing pass</u>~~said step of loading the log~~

~~comprises the steps of:~~

~~———reading the log records from said one or more log disks; and~~

~~———playing the log records in two pass to restore the main memory database to the latest~~

~~consistent state.~~

11.   (Currently Amended) The method of Claim 10, wherein ~~the step of~~ reading the log

records and ~~the step of~~ <u>re</u>playing the log records are executed in a pipeline.

12.   (Currently Amended) The method of Claim 10, wherein ~~the step of~~ reading the log

records is executed in parallel by partitioning the log records as well as ~~the step of~~ <u>re</u>playing the

log records.

13.   (Currently Amended) The method of Claim 12, wherein ~~the step of~~ reading the log

records and ~~the step of~~ <u>re</u>playing the log records are executed in a pipeline.

14.   (Currently Amended) The method of Claim 8, wherein ~~the step of~~ <u>replaying the log</u>

<u>records is done in one pass</u>~~loading the log comprises the steps of:~~

~~reading log records from said one or more log disks; and~~

~~playing the log records in one pass to restore the main memory database to the latest~~

~~consistent state~~.

15.   (Currently Amended) The method of Claim 14, wherein ~~the step of~~ reading the log

records and ~~the step of~~ <u>re</u>playing the log records are executed in a pipeline.

16.   (Currently Amended) The method of Claim 14, wherein ~~the step of~~ reading the log

records <u>and replaying the log records are</u> ~~is~~ executed in parallel by partitioning the log records ~~as~~

~~well as the step of playing the log records~~.

11

17.   (Currently Amended) The method of Claim 16, wherein ~~the step of~~ reading the log

records and ~~the step of~~ replaying the log records are executed in a pipeline.

18.   (Currently Amended) The method of Claim 8, further comprising ~~the step of~~ filling the

main memory database with 0s in advance.

19.   (Currently Amended) The method of Claim 18, wherein ~~the step of~~ loading the ~~backup~~

~~data~~ <u>checkpointed database</u> comprises ~~the steps of~~:

reading the ~~backup data~~ <u>checkpointed database</u> from said one or more ~~backup~~

~~disks~~ <u>backup storage devices</u>; and

playing the ~~backup data~~ <u>checkpointed database</u> by applying the XOR operation between

the ~~backup data~~ <u>checkpointed database</u> and the main memory database.

20.   (Currently Amended) The method of Claim 19, wherein ~~the step of~~ reading the

~~backup data~~ <u>checkpointed database</u> and ~~the step of~~ playing the ~~backup data~~ <u>checkpointed</u>

<u>database</u> are executed in a pipeline.

21.   (Currently Amended) The method of Claim 19, wherein ~~the step of~~ reading the ~~backup~~

~~data~~ <u>checkpointed database</u> is executed in parallel by partitioning the ~~backup data~~ <u>checkpointed</u>

database as well as ~~the step of~~ playing the ~~backup data~~<u>checkpointed database</u>.

22.   (Currently Amended) The method of Claim 21, wherein ~~the step of~~ reading the ~~backup~~

~~data~~<u>checkpointed database</u> and ~~the step of~~ playing the ~~backup data~~<u>checkpointed database</u> are

executed in a pipeline.

23.   (Currently Amended) The method of Claim 19, wherein ~~the step of~~ loading the ~~backup~~

~~data~~<u>checkpointed database</u> and ~~the step of~~ loading the log records are executed in parallel.

24.   (Currently Amended) A transaction processing system allowing <u>logging updates and</u>

recovery from a failure, comprising<u>:</u>

<u>a</u> main memory for storing a database~~;~~<u>;</u>

~~one or more~~<u>a plurality of persistent log storage devices</u> ~~log disks~~ for storing log records

for <u>parallel logging and</u> parallel recovery ~~of the main memory database,~~<u>;</u>

one or more ~~backup disks~~<u>persistent backup storage devices</u> for storing ~~a copy of~~ the

database <u>in the</u> main memory ~~database~~;

means for generating ~~a~~ differential log <u>records</u> ~~as part of the log body~~ by applying a bit-

wise exclusive-OR (XOR) between a before-<u>update</u> image <u>an after-update image;</u>~~of the database~~

~~before an update to the database is made and an after-image of the database after the update is~~

13

Serial No.   09/768,301   Docket No.   K-0254
Amdt. dated September 26, 2003
Reply to Office Action of <u>March 26, 2003</u>

~~made; and~~

means for distributing the generated differential log records in parallel to said persistent

log storage devices; and ~~means for recovering from a failure by applying the XOR operation~~

~~between the differential log and the before-image.~~

means for replaying the differential log records in an arbitrary order, independent of

their generation order, by using the bit-wise XOR operations.

25.   (Original) The system of Claim 24, wherein the database comprises a plurality of fixed-

size pages.

26.   (Currently Amended) The system of Claim 24, further comprising:

means for checkpointing <u>the database</u> by occasionally writing the database in the main

memory to one or more <u>persistent backup storage devices.</u> ~~backup disks as backup data; and~~

27.   (Original) The system of Claim 26, wherein the means for checkpointing uses the

transaction consistent checkpointing policy.

28.   (Original) The system of Claim 26, wherein the means for checkpointing uses the

action consistent checkpointing policy.

14

29.   (Original) The system of Claim 26, wherein the means for checkpointing uses the fuzzy

checkpointing policy.

30.   (Currently Amended) The system of Claim 26, wherein the means for recovering

comprises:

        means for loading the ~~backup data~~<u>checkpointed database</u> into the main memory

database; and

        means for loading the log into the main memory database.

        ~~[system + checkpointing + BL/LL + BR/BP]~~

31.   (Currently Amended) The system of Claim 30, wherein the means for loading the

~~backup data~~<u>checkpointed database</u> comprises:

        means for reading the ~~backup data~~<u>checkpointed database</u> from one or more ~~backup~~

~~disk~~<u>persistent backup storage device</u>~~s~~; and

        means for playing the ~~backup data~~<u>checkpointed database</u> to restore the main memory

_database to the state when the backup was made by applying the XOR operation between the

~~backup data~~<u>checkpointed database</u> and the main memory database.

32.   (Currently Amended)   The system of Claim 30, wherein the means for loading the log

15

comprises:

      means for reading the log records from the <u>persistent log storage devices</u>~~log disk~~; and

      means for playing the log records in two pass to restore the main memory database to the latest consistent state.

33.   (~~Original~~<u>Currently Amended</u>) The system of Claim 30, wherein the means for loading the log comprises:

      means for reading the log records from the <u>persistent log storage devices</u>~~log disk~~; and

      means for playing the log records in one pass to restore the main memory database to the latest consistent state.

34.   (Currently Amended) A computer-readable storage medium that contains a program for logging updates <u>and recovering from failure</u> in a ~~main-memory~~ transaction-processing system having <u>a</u> main memory for storing a database, one or more ~~log-disks~~<u>persistent backup storage devices</u> for storing <u>the database in the</u> ~~log records for parallel recovery of the~~ main memory ~~database~~, and one or more ~~backup-disks~~<u>persistent log storage devices</u> for storing <u>log records for parallel logging and parallel recovery</u> ~~a copy of the main memory database~~, where the program under the control of a CPU performs the steps of:

     ~~taking a before image of the database before an update to the database is made;~~

     ~~taking an after image of the database after the update is made;~~

generating ~~a~~ differential log <u>records</u> ~~as a log body of each log record~~ by applying ~~a~~ bit-wise exclusive-OR (XOR) operation<u>s</u> between ~~the before-image and the after-image~~<u>before-update images and after-update images</u>;

<u>distributing the generated differential log records in parallel to said persistent log storage devices; and</u>~~recovering from a failure by applying the XOR operation between the differential log and the before-image.~~

<u>replaying the differential log records in an arbitrary order, independent of their generation order, by using the bit-wise XOR operations.</u>

35.   (Original) The storage medium of Claim 34, wherein the medium is a CD.

~~37.~~<u>36.</u>   (Currently Amended) The storage medium of Claim 34, wherein the medium is a magnetic tape.

37.   (New) The method of Claim 1, further comprising one or more in-memory log buffers wherein each generated log record is temporarily stored in any available log buffer and a group of the buffered log records are written together to an arbitrary one of said one or more persistent log storage devices.